UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/717,674	11/21/2000	Steven J. Kruy	777.360US1	9105

41505 7590 11/16/2004

WOODCOCK WASHBURN LLP
ONE LIBERTY PLACE - 46TH FLOOR
PHILADELPHIA, PA 19103

EXAMINER	
KLINGER, SCOTT M	

ART UNIT	PAPER NUMBER
2153	7

DATE MAILED: 11/16/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/717,674	KRUY ET AL.	
	Examiner	Art Unit	
	Scott M. Klinger	2153	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 November 2000*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claims 1-39 are pending.

Priority

No claim for priority has been made. The effective filing date for subject matter in the application is 21 November 2000.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 16 and 31 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 16 recites the limitation "the timer value" on page 28, line 23. There is insufficient antecedent basis for this limitation in the claim.

Claim 31 recites the limitation "the server" on page 31, line 18. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted

on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1, 2, 5-9, 11, 19-23, 26-29, 33-36, and 39 are rejected under 35 U.S.C. 102(e) as being anticipated by Chorn (U.S. Patent Number 6,275,843, hereinafter "Chorn"). Chorn discloses a method and apparatus for processing multiple service requests within a global transaction by a single server application program instance. Chorn shows,

In referring to claim 1,

- An application interface of a client computer receiving a request from an application program:

  Chorn, Fig. **1** shows an application interface made up of a resource manager **16**, a transaction manager **18**, and a communication resource manager **20**, which receives requests from application program **14**
- The application interface associating a transaction identifier with the request, wherein the transaction identifier identifies a transaction that the request is associated with:

  "The APs begin and end transactions under the control of a Transaction Manager (TM) 18. The TM is a system software component that assigns transaction identifiers to global transactions, monitors their progress, coordinates their completion, and coordinates failure recovery. The TM enforces the transaction property of atomicity. If a global transaction is being processed, the TM adheres to the two-phase Commit transaction processing protocol." (Chorn, col. 7, lines 13-20)
- The application interface forming a message including the transaction identifier and the request; the application interface sending the message to a server; the server receiving the

message; the server processing the request in a context of the transaction identified by the transaction identifier in the message:

Chorn, Fig. **4** shows forming and sending a message including the transaction identifier:

"Thread 102 represents the initial processing performed by the Client System 10 in setting up to process a global transaction. The OSI-TP Thread 104 is initiated with Service Request-1. Data transfer Line 106 shows Service Request-1 being transferred from the Client System to the Server System 12 ... Thread 120 then waits for further transaction control messages (Prepare, Commit, rollback) associated with Service Request-1 from the Client System. Reply-1 and control are returned to Thread 128." (Chorn, col. 9, line 55 – col. 10, line 18)

In referring to claim 2,

- The application interface associating a sequence indicator to the request, wherein the sequence indicator indicates in what sequence the server should process the request within the context of the transaction; and wherein forming the message comprises including the sequence indicator in the message:

  Chorn, Fig. **4** shows service requests having sequence indicators

In referring to claim 5,

- The server allocating a database connection to the transaction; and the server processing the requests that form a part of the transaction over the database connection allocated to the transaction:

  "A global transaction consists of multiple, coordinated database updates, possibly occurring on different computers. Global transactions are used when it is important that all databases are synchronized so that either all updates are made or none are made." (Chorn, col. 2, lines 45-49), a system in which a server connects to a database inherently implies allocating a connection to said database

In referring to claim 6,

- The application interface including a sequence indicator in the message, wherein the sequence indicator indicates in what sequence the server should process the request within the context of the transaction; and wherein processing the requests comprises processing the requests in an order indicated by the sequence indicator:

  Chorn, Fig. **4** shows service requests having sequence indicators and processing said request in the order of the sequence indicators

In referring to claim 7,

- Receiving a request from an application program:

  Chorn, Fig. **1** shows an application interface made up of a resource manager **16**, a transaction manager **18**, and a communication resource manager **20**, which receives requests from application program **14**
- Associating a transaction identifier with the request, wherein the transaction identifier identifies a transaction that the request is associated with:

  Chorn, col. 7, lines 13-20 (see full quote above)
- Forming a message including the transaction identifier and the request; and sending the message to a server.

  Chorn, Fig. **4** shows forming a message including the transaction identifier and the request; and sending the message to a server

In referring to claim 8,

- Assigning the transaction identifier to the transaction if the request is a request to open a new transaction:

  Chorn, col. 7, lines 13-20 (see full quote above), a transaction identifier is associated with all of the requests

In referring to claim 9,

- Associating a sequence indicator with the request, wherein the sequence indicator indicates in what sequence the server should process the request within the context of the transaction; and wherein forming the message comprises including the sequence indicator within the message:

  Chorn, Fig. **4** shows service requests having sequence indicators and processing said request in the order of the sequence indicators

In referring to claim 11,

- Receiving a reply from the server in response to sending the message; and sending the reply to the application program:

  Chorn, col. 9, line 55 – col. 10, line 18 (see full quote above)

In referring to claim 19,

- Receiving a message from a client:

  Chorn, Fig. **1** shows an application interface made up of a resource manager **16**, a transaction manager **18**, and a communication resource manager **20**, which receives requests from application program **14**

- The message includes a transaction identifier that indicates that a request specified in the message should be performed in a context of a transaction:

  Chorn, col. 7, lines 13-20 (see full quote above)

- Processing the request in the context of the transaction identified by the transaction identifier:

  Chorn, Fig. **4** shows forming a message including the transaction identifier and the request; and sending the message to a server

In referring to claim 20,

- The message includes a sequence indicator which indicates a sequence, within the transaction, that the request should be processed, and wherein processing the request comprises processing the request in the sequence:

  Chorn, Fig. **4** shows service requests having sequence indicators and processing said request in the order of the sequence indicators

In referring to claim 21,

- Sequencing the request by placing the request, and other requests that comprise the transaction, in a transaction queue associated with the transaction; and wherein processing the request comprises processing the request in a sequence of requests in the transaction queue:

  Chorn, Fig. **4** shows service requests having sequence indicators and processing said request in the order of the sequence indicators. A client-server system implies queues on either end to buffer received packets.

In referring to claim 22,

- Reserving a database connection for the transaction between the server and a database corresponding to the transaction:

  Chorn, col. 2, lines 45-49 (see full quote above), a system in which a server connects to a database inherently implies allocating a connection to said database

In referring to claim 23,

- Determining whether a free connection to the database is available; and if the free connection is available, reserving the database connection by mapping the transaction identifier to the free connection:

  Chorn, col. 2, lines 45-49 (see full quote above), a system in which a server connects to a database inherently implies determining if said database is free

In referring to claim 26,

- Closing the connection when the transaction is committed or aborted:
  Chorn Fig. 4 shows the server sends a done message when the transaction is completed effectively closing the connection

In referring to claim 27,

- Deallocating the connection when the transaction is committed or aborted:
  Chorn Fig. 4 shows the server sends a done message when the transaction is completed effectively deallocating the connection

In referring to claim 28,

- A processor for executing an application interface which receives requests destined for the server from at least one application program:
  Chorn, Fig. **1** shows an application interface made up of a resource manager **16**, a transaction manager **18**, and a communication resource manager **20**, which receives requests from application program **14**, a processor for executing an application is inherently implied in a client that executes an applicaiton
- Associating transaction identifiers with each of the requests that are associated with transactions:
  Chorn, col. 7, lines 13-20 (see full quote above)
- Forming messages including the transaction identifiers and the requests:
  Chorn, Fig. **4** shows forming a message including the transaction identifier and the request; and sending the message to a server
- An interface to the server, coupled to the processor through a bus, which sends the messages to the server:
  Chorn, Fig. **1** shows an interface to the server **21**, a processor being connected to said interface is inherently implied in a system that executes an application program and sends data over an interface

In referring to claim 29,

- The processor is further for associating sequence indicators with the requests, wherein the sequence indicators indicate the order that the requests within a transaction should be performed by the server:

  Chorn, Fig. **4** shows service requests having sequence indicators and processing said request in the order of the sequence indicators

In referring to claim 33,

- A processor for executing a transaction manager that receives, via a network interface, a message from a client:

  Chorn, Fig. **1** shows a network interface **21**, a processor being connected to said interface is inherently implied in a system that executes a server application program and sends/receives data over an interface
- The message includes a transaction identifier that indicates that a request specified in the message should be performed in a context of a transaction, and for processing the request in the context of the transaction identified by the transaction identifier; and the network interface for receiving the message from the client:

  Chorn, col. 9, line 55 – col. 10, line 18 (see full quote above)

In referring to claim 34,

- The message also includes a sequence indicator, which indicates a sequence, within the transaction, that the request should be processed, and wherein the transaction manager processes the request in the sequence:

  Chorn, Fig. **4** shows service requests having sequence indicators and processing said request in the order of the sequence indicators

In referring to claim 35,

- The transaction manager also reserves a database connection for the transaction between the server and a database corresponding to the transaction:
  Chorn, col. 2, lines 45-49 (see full quote above), a system in which a server connects to a database inherently implies allocating a connection to said database

In referring to claim 36,

- Receiving a request from an application program; associating a transaction identifier with the request, wherein the transaction identifier identifies a transaction that the request is associated with; Forming a message including the transaction identifier and the request; and sending the message to a server:
  Chorn, col. 9, line 55 – col. 10, line 18 (see full quote above)

In referring to claim 39,

- Receiving a message from a client, wherein the message includes a transaction identifier that indicates that a request specified in the message should be performed in a context of a transaction; processing the request in the context of the transaction identified by the transaction identifier:
  Chorn, col. 9, line 55 – col. 10, line 18 (see full quote above)

Claims 12, 30, and 37 are rejected under 35 U.S.C. 102(e) as being anticipated by Bergadano (U.S. Patent Number 6,574,627, hereinafter "Bergadano"). Bergadano discloses a method and apparatus for the verification of server access logs and statistics. Bergadano shows,

In referring to claim 12,

- Receiving a request from an application program, wherein the application interface is separate from the application program; formatting the request into a message; sending the

message to the first server; receiving a reply from the first server in response to the message, wherein the reply includes a redirect request that indicates that the second server is a correct destination for the message; and

"When B=1, the client connects to the audit server because it receives a redirect request from the application server (phase 205 in FIG. 2). As is shown in FIG. 3B, the audit server 103 redirects the client 101 again to the application server 100. Moreover, the audit server logs the request for later comparison with the application server log files. As result, after this second redirection, the client connects again to the application server first contacted and issues its request again. Referring now to FIG. 3B and FIG. 2, phase 300 of FIG. 3B corresponds to phases 200 and 201 of FIG. 2, phase 301 corresponds to phase 202, phase 304 corresponds to phases 203 and 204 (with B=1), and phase 305 corresponds to phase 205. Phases 306 and 307 will be transparent for the user as long as the client is protocol compliant." (Bergadano, col. 7, line 39-52)

- Sending the message to the second server without involving the application program.

  "Phases 306 and 307 will be transparent for the user as long as the client is protocol compliant." (Bergadano, col. 7, line 51-52)

In referring to claim 30,

- A processor for executing an application interface and a network interface, coupled to the processor through a bus, which sends the message to the first server and the second server:

  "When a client requests a document, it uses the relevant application protocol to send its request" (Bergadano, col. 3, line 60-61)

- Receiving a request destined for a first server from an application program, formatting the request into a message, and sending the message to the first server via a network interface; receiving a reply from the first server in response to the message, wherein the reply includes a redirect request that indicates that a second server is a correct destination for the message:

  Bergadano, col. 7, line 39-52 (see full quote above)

- Sending the message to the second server via the network interface without involving the application program:
  Bergadano, col. 7, line 51-52 (see full quote above)

In referring to claim 37,

- Receiving a request from an application program, wherein the application interface is separate from the application program; formatting the request into a message; sending the message to the first server; receiving a reply from the first server in response to the message, wherein the reply includes a redirect request that indicates that the second server is a correct destination for the message:
  Bergadano, col. 7, line 39-52 (see full quote above)
- Sending the message to the second server without involving the application program:
  Bergadano, col. 7, line 51-52 (see full quote above)

Claims 15-18, 31, 32, and 38 are rejected under 35 U.S.C. 102(b) as being anticipated by RFC 793 (RFC 793, September 1981). RFC 793 discloses Transport Control Protocol. RFC 793 shows,

In referring to claim 15,

- Opening a connection with the server; sending the one or more message over the connection:
  "The TCP is intended to provide a reliable process-to-process communication service in a multinetwork environment. The TCP is intended to be a host-to-host protocol in common use in multiple networks." (RFC 793, page 2)
  "A transmission control block (TCB) is created and partially filled in with data from the OPEN command parameters. On an active OPEN command, the TCP will begin the procedure to synchronize (i.e., establish) the connection at once." (RFC 793, page 45)

- Determining whether a timeout period has expired; and closing the connection when the timeout period has expired:

  "The timeout, if present, permits the caller to set up a timeout for all data submitted to TCP. If data is not successfully delivered to the destination within the timeout period, the TCP will abort the connection. The present global default is five minutes." (RFC 793, page 45)

In referring to claim 16,

- Initializing and starting a timer when a reply is received from the server; and wherein determining whether the timeout period has expired comprises comparing the timer value to the timeout period:

  "The timeout, if present, permits the caller to set up a timeout for all data submitted to TCP. If data is not successfully delivered to the destination within the timeout period, the TCP will abort the connection. The present global default is five minutes." (RFC 793, page 45)

In referring to claim 17,

- Receiving another request destined for the server within the timeout period; and sending the another request to the server over the connection:

  TCP allows requests to be sent before acknowledgement of previous requests have been received

In referring to claim 18,

- Re-initializing and re-starting the timer when a reply corresponding to the another request is received from the server:

  The connection timeout is reset every time data is recieved

In referring to claim 31,

- Opens a connection with the server, sends the one or more requests over the connection,:
  "The TCP is intended to provide a reliable process-to-process communication service in a multinetwork environment. The TCP is intended to be a host-to-host protocol in common use in multiple networks." (RFC 793, page 2)
- Determines whether a timeout period has expired, and closes the connection when the timeout period has expired:
  "The timeout, if present, permits the caller to set up a timeout for all data submitted to TCP. If data is not successfully delivered to the destination within the timeout period, the TCP will abort the connection. The present global default is five minutes." (RFC 793, page 45)
- A processor for executing an application interface which receives requests destined for the server from at least one application program, an interface to the server, coupled to the processor through a bus, which sends the one or more requests to the server:
  A system that provides a process-to-process communication service in a multinetwork environment inherently implies a processor for executing an application interface which receives requests destined for the server from at least one application program, an interface to the server, coupled to the processor through a bus, which sends the one or more requests to the server

In referring to claim 32,

- The processor is further for sending one or more additional requests destined for the server over the connection during the timeout period.
  "The interface between TCP and lower level protocol is essentially unspecified except that it is assumed there is a mechanism whereby the two levels can asynchronously pass information to each other." (RFC 793, page 3), a system that sends data asynchronously implies sending requests during the timeout period

In referring to claim 38,

- Opening a connection with the server; sending the one or more messages over the connection:

  "The TCP is intended to provide a reliable process-to-process communication service in a multinetwork environment. The TCP is intended to be a host-to-host protocol in common use in multiple networks." (RFC 793, page 2)

- Determining whether a timeout period has expired; and closing the connection when the timeout period has expired:

  "The timeout, if present, permits the caller to set up a timeout for all data submitted to TCP. If data is not successfully delivered to the destination within the timeout period, the TCP will abort the connection. The present global default is five minutes." (RFC 793, page 45)

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chorn in view of RFC 793 (RFC 793, September 1981). Although Chorn shows substantial features of the claimed invention, Chorn does not explicitly show a timeout. Nonetheless this feature is well known in the art and would have been an obvious implementation of the system disclosed by Chorn as evidenced by RFC 793.

In analogous art, RFC 793 discloses Transport Control Protocol. RFC 793shows:

- Opening a connection with the server; sending the message over the connection;

"The TCP is intended to provide a reliable process-to-process communication service in a multinetwork environment. The TCP is intended to be a host-to-host protocol in common use in multiple networks." (RFC 793, page 2)

"A transmission control block (TCB) is created and partially filled in with data from the OPEN command parameters. On an active OPEN command, the TCP will begin the procedure to synchronize (i.e., establish) the connection at once." (RFC 793, page 45)

- Determining whether a timeout period has expired; and closing the connection when the timeout period has expired:

  "The timeout, if present, permits the caller to set up a timeout for all data submitted to TCP. If data is not successfully delivered to the destination within the timeout period, the TCP will abort the connection. The present global default is five minutes." (RFC 793, page 45)

Given these teachings, a person of ordinary skill in the art would have readily recognized the desirability and advantages of implementing the system of Chorn so as to use TCP/IP, such as taught by RFC 793, in order to communicate from clients to servers over the Internet.

Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chorn in view of Bergadano. Although Chorn shows substantial features of the claimed invention, Chorn does not show redirecting the client to a second server. Nonetheless this feature is well known in the art and would have been an obvious modification to the system disclosed by Chorn as evidenced by Bergadano.

In analogous art, Bergadano discloses a method and apparatus for the verification of server access logs and statistics. Bergadano shows receiving a reply from the server in response to the request; determining that the reply includes a redirect request that indicates that the request should be sent to another server; and sending the message to the another server: *Bergadano, col. 7, line 39-52* (see full quote above).

Given these teachings, a person of ordinary skill in the art would have readily recognized the desirability and advantages of modifying the system of Chorn so as to redirecting the client to a

second server, such as taught by Bergadano, in order to authenticate the client without user intervention or awareness.

Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chorn in view of Valiant (U.S. Patent Number 5,608,870, hereinafter "Valiant"). Although Chorn shows substantial features of the claimed invention, Chorn does not show combining a plurality of requests into a single combined request. Nonetheless this feature is well known in the art and would have been an obvious modification to the system disclosed by Chorn as evidenced by Valiant.

In analogous art, Valiant discloses system for combining a plurality of requests referencing a common target address into a single combined request having a single reference to the target address. Valiant shows: *"A goal of the invention is to provide an efficacious method of providing a multiple processor computer with a combining facility that is acceptably efficient for the widest range of concurrent access patterns, but that does not require a hardware combining network. The method is universal, that is, it operates with no programmer intervention other than possibly the setting of a small number of parameters."* (Valiant, col. 1, lines 57-63).

Given these teachings, a person of ordinary skill in the art would have readily recognized the desirability and advantages of modifying the system of Chorn so as to combine a plurality of requests into a single combined request, such as taught by Valiant, in order to reduce overhead.

Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bergadano in view of RFC 793. Although Bergadano shows substantial features of the claimed invention, Bergadano does not explicitly show a timeout. Nonetheless this feature is well known in the art and would have been an obvious implementation of the system disclosed by Bergadano as evidenced by RFC 793.

In analogous art, RFC 793 discloses Transport Control Protocol. RFC 793shows:

- Opening a connection with the server; sending the message over the connection;

"The TCP is intended to provide a reliable process-to-process communication service in a multinetwork environment. The TCP is intended to be a host-to-host protocol in common use in multiple networks." (RFC 793, page 2)

"A transmission control block (TCB) is created and partially filled in with data from the OPEN command parameters. On an active OPEN command, the TCP will begin the procedure to synchronize (i.e., establish) the connection at once." (RFC 793, page 45)

- Determining whether a timeout period has expired; and closing the connection when the timeout period has expired:

  "The timeout, if present, permits the caller to set up a timeout for all data submitted to TCP. If data is not successfully delivered to the destination within the timeout period, the TCP will abort the connection. The present global default is five minutes." (RFC 793, page 45)

Given these teachings, a person of ordinary skill in the art would have readily recognized the desirability and advantages of implementing the system of Bergadano so as to use TCP/IP, such as taught by RFC 793, in order to communicate from clients to servers over the Internet.

Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bergadano. Although Bergadano shows substantial features of the claimed invention, Bergadano does not show caching redirect requests. Nonetheless this feature is well known in the art and would have been an obvious modification to the system disclosed by Bergadano.

Bergadano discloses a system that sends redirect requests for services that are not provided by the server that received a request. A person of ordinary skill in the art would have readily recognized the desirability and advantages of modifying the system of Bergadano so as to cache redirect requests, in order to decrease the bandwidth used by the original request and the redirect request.

Claims 24 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chorn. Although Chorn shows substantial features of the claimed invention, Chorn is silent as to what happens when connections to databases are not available. Chorn does not explicitly show reserving the database connection, determining if a maximum number of connections has been made to the database or stalling the request until the database is available. Nonetheless this feature is well known in the art and would have been an obvious implementation of the system disclosed by Chorn.

The system of Chorn involves coordinating database updates: *"A global transaction consists of multiple, coordinated database updates, possibly occurring on different computers. Global transactions are used when it is important that all databases are synchronized so that either all updates are made or none are made."* (Chorn, col. 2, lines 45-49)

A person of ordinary skill in the art would have readily recognized the desirability and advantages of implementing the system of Chorn so as to reserve a database connection, in order to complete the database transaction when the database is no longer busy.

A person of ordinary skill in the art would have readily recognized the desirability and advantages of implementing the system of Chorn so as to determine whether a maximum number of database connections had been reached and stalling the requests to said database, in order to complete the database transactions when the database is no longer busy.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Scott M. Klinger whose telephone number is (703) 305-8285. The examiner can normally be reached on M-F 7:00am - 3:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenn Burgess can be reached on (703) 305-4792. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Scott M. Klinger
Examiner
Art Unit 2153

smk

Bradley Edelman
ART UNIT 2153